

## A PACKET SWITCHED COMMUNICATIONS SYSTEM FOR GRO

Shabu Husain, Wen Hsing Yang and Rani Vadlamudi

AlliedSignal Technical Services Corp.

7515 Mission Drive

Seabrook, MD 20706

Joseph Valenti

Goddard Space Flight Center

Greenbelt, MD 20771

### Abstract

This paper describes the packet switched Instrumenters Communication System (ICS) that has been developed for the Command Management Facility at Goddard Space Flight Center (GSFC) to support the Gamma Ray Observatory (GRO) spacecraft. The GRO ICS serves as a vital science data acquisition link to the GRO scientists to initiate commands for their spacecraft instruments. The system is ready to send and receive messages at any time, 24 hours a day and seven days a week. The system is based on X.25 and the International Standard Organization's (ISO) 7-layer Open Systems Interconnection (OSI) protocol model and has client and server components. The components of the GRO ICS are discussed along with how the Communications Subsystem for Interconnection (CSFI) and Network Control Program Packet Switching Interface (NPSI) software are used in the system.

### 1. INTRODUCTION

The Command Management System (CMS) at NASA-GSFC supports scientific experiments by providing for the planned, safe operation of scientific spacecraft. CMS software is responsible for command request processing, command load generation and checking, constraint checking, automatic command sequence generation, and onboard computer memory management for the spacecraft. One of the primary ground interfaces supported by the CMS is an electronic interface with the remotely-located GRO Instrumenters (NASA/GSFC, [1986]) that allows them to take science data acquisition requests from GRO scientists and translate the requests into command requests. The interface is

essential for GRO science data acquisition as well as the safe and effective control of the GRO spacecraft.

The GRO ICS is based on the International Standard Organization's 7-layer Open Systems Interconnection (OSI) protocol model. A schematic representation of the ISO-OSI model is shown in Figure 1.

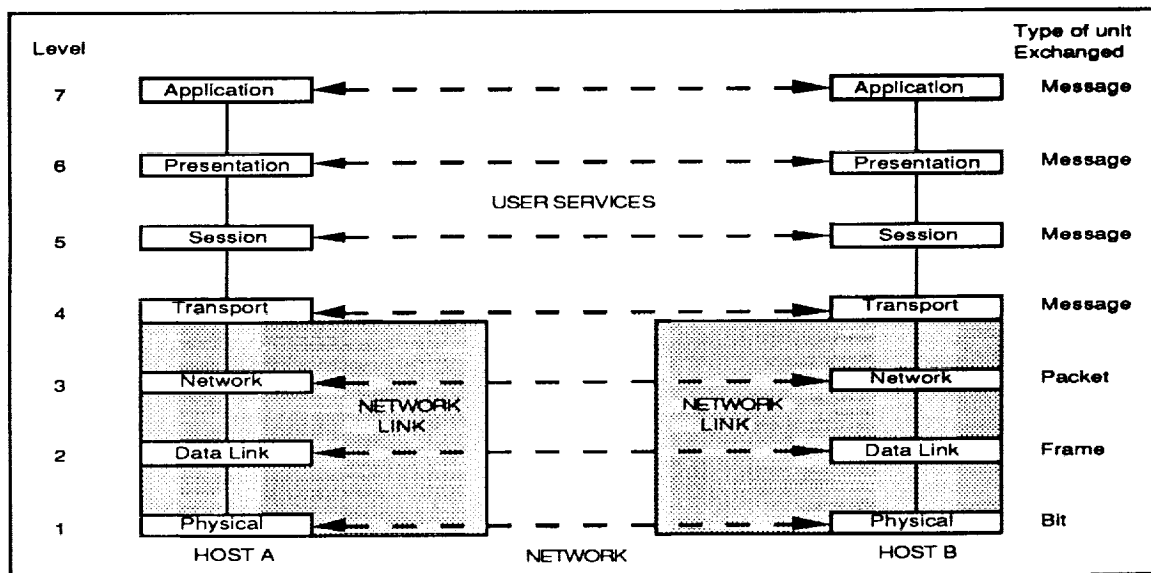


Figure 1. Representation of the ISO-OSI Model for Network Architecture

Each level within the OSI model performs a specific function while allowing the systems being connected to have a heterogeneous nature. The functions of levels 1 to 3 of the model deal with the internal mechanisms of the network and their interfaces to the hosts. This is significant for X.25 because the highest level of protocol that X.25 defines is the packet level (or X.25 at Level 3) which takes care of all OSI network (Level 3) layer functions and some transport layer (Level 4) functions. Levels 4 and above deal with protocols for controlling processes on the hosts themselves.

The first three (bottom) levels of the GRO ICS were implemented using X.25. These levels control the exchange of data between the user devices and the packet network node of the packet switching network. At the physical layer (layer 1), the NASA Communication (NASCOM) Division provides dedicated communication services to accommodate data transfer between the CMS and the users. The top four layers were developed and integrated for the GRO ICS in accordance with the ISO model recommendations.

The GRO ICS has been designed and built with the client/server concept to provide each of the four remotely-located GRO Instrumenters with an X.25 interface that is able to send and receive messages at any time, 24 hours a day and seven days a week. For incoming messages from any GRO Instrumenter the ICS functions as a server to store and forward the messages to the CMS for further processing. For outgoing messages from the CMS to the investigators, the ICS functions as a client to establish a communication link to the designated instrumenter's communication server task. In addition to client and server components, the GRO ICS includes the Application Programming Interface (API) component and the Network Subsystem Software (NSS) component. These components are shown schematically in Figure 2. Each of the components is described in more detail in the following sections of this paper.

## **2. THE SERVER**

The GRO ICS has four identical servers, one for each GRO Instrumenter, and each server has a subserver. Each server controls the attaching, detaching and scheduling tasks of the GRO ICS and maintains the sessions which are in progress. Certain tasks are attached by the server at the start up time while others are attached as they are required. At the start up, the server task attaches LSTNR1S, LSTNR2S and one standby receiver task (SUBSRV01). If the server receives a ring request from an investigator, it posts the standby receiver task (SUBSRV01) and

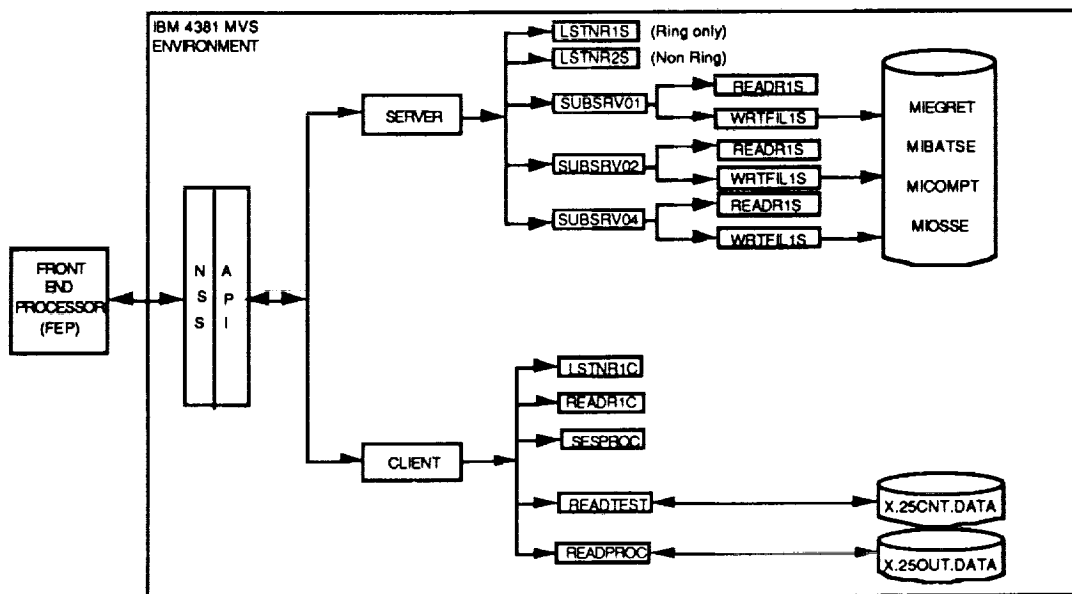


Figure 2. GRO ICS Structure Diagram

attaches a new standby receiver task (SUBSRV02). For every new ring request the standby receiver task is posted and a new receiver task is attached in the standby mode. This procedure is faster than attaching a receiver task after the receipt of a ring request.

The LSTNR1S and LSTNR2S tasks are similar to each other, except that they listen for different types of messages. LSTNR1S listens for supervisory messages (ring-only requests) coming over LCN0. At the same time, LSTNR2S listens for all non-supervisory messages on LCN0. LSTNR1S is scheduled to be executed at the highest priority among all the subtasks of the server, which ensures the quickest possible reply to incoming ring requests.

As soon as a message arrives at either the LSTNR1S or LSTNR2S tasks, the server task is posted and the content of the message is deciphered to determine whether it is Logical Channel Number (LCN) oriented or a non-LCN oriented. If the message is LCN oriented, the server determines which of the attached tasks is assigned to that particular LCN. If the message is non-LCN

oriented, the server posts each and every read task as well as the write task. Each task will in turn process the message.

After determining the responsible task for a supervisory message, the server posts the task and action is taken by either the READR1S or the WRTFIL1S task. The function of READR1S task is to read all the data coming from the FEP for the specified LCN value. The function of the WRTFIL1S task is to write the incoming data messages to one of the four user files (MIEGRET, MIBATSE, MICOMPTTEL or MIOSSE), depending on who initiated that session. WRTFIL1S is created so that the processes of writing data to a file and receiving data messages can be done asynchronously.

### **3. THE CLIENT**

The client (sender) task has some similarity to the receiver task (SUBSRV). The client attaches two tasks (LSTNR1C and READR1C) during initialization. The function of the LSTNR1C task is to listen for incoming replies over the specified LCN from the FEP. The SESPROC task builds the session control blocks as the session handshaking requests are received from the session control file (X25CNT.DATA). The client then initiates a session by sending a connect request for an available LCN and a ring request to the appropriate remote host. After a ring has been accepted, the client initiates the session initiation handshake. If the above steps are successful, the client sends the message from the output queue (X25OUT.DATA) to the appropriate remotely-located instrumenter.

After successfully sending data to the remotely-located instrumenter, a session termination request is sent to the host followed by a clear request to terminate the connection. The client task will perform these steps every time an automatic or manual session handshaking request is received from the operator. The advantage of initiating an automatic session handshake is

that sending and receiving session control blocks and transmitting data is accomplished without any delays or session sequencing errors. When there is no communication activity, the client can be terminated gracefully so that only the server's listener task is active. With this design for the client component, the overhead on the GRO ICS operating system is significantly reduced.

#### **4. THE APPLICATION PROGRAMMING INTERFACE (API)**

The API provides users with access to the X.25 networking capability and the API application programs that are running under the IBM MVS operating system. Several C-language application programs were written for the API to access the NSS.

#### **5. THE NETWORK SUBSYSTEM SOFTWARE (NSS)**

The NSS acts as a message processor for requests for network services from the API application programs. The GRO Instrumenters are able to establish connections, send and receive data, and terminate connections via requests that are passed by the NSS to the network. The NSS multiplexes requests from users through I/O sub channels while it manages all of the queues and control blocks necessary to support network traffic. The NSS is interfaced to the Comten 3695 Front End Processor (FEP) which has an X.25 interface to the IBM 4381 mainframe that supports the CMS.

#### **6. CSFI AND NPSI SOFTWARE**

The GRO ICS currently uses IBM Communications Subsystem for Interconnection (CSFI) software. CSFI is a communication and transmission control program for networking IBM computers with non-SNA hosts through an X.25 network. CSFI was developed by IBM in France as the Generalized Transaction Monitor for Open System Interconnection (GTMOSI) software. The product became available

as CSFI in Europe and Canada in April, 1990. GSFC was able to obtain CSFI in Dec. 1990, a month before CSFI was officially released in the U.S. This allowed the GRO ICS to be the first system in the U.S. to use CSFI software under the MVS operating system.

CSFI allows state-of-the-art communication system applications to be written through a programming interface. The programming interface consists of a set of macro statements that can be invoked in an assembler language program. CSFI also provides ready-made services for frequently used types of connections between heterogeneous elements. Some of these services may be customized to a user's unique requirements. In addition, CSFI provides the user with the ability to process X.25 call packets.

When a call packet is received from a GRO Instrumenter, CSFI sends out a call accept packet and starts the server task to receive data. The server calls a subserver that creates a queue to store any data received from the instrumenter. A timer is started and the server waits for a message from the instrumenter. If a session initiation request is received, the server sends a session initiation acceptance message and waits for the next message. If a message initiation request is received, the server sends a message initiation acceptance message and waits for the next message. If data is received, it is written to the queue and the server waits for another message. If a message termination request is received, the server sends a message termination confirmation message and waits for the next message. If a session termination request is received, the server sends a session termination confirmation message, waits for the primary to terminate, and passes control to the subserver.

When there is data, a subserver is called by the server. A sequential queue is created to receive the data, control is passed back to the server, and the server waits for the subserver to terminate. Once the subserver has terminated, an entry from

the queue is read. If the entry is a message header, bytes 4 through 32 are translated to EBCDIC and the remaining part of the buffer is filled with null characters. The message name is retrieved from the header, and the header is written to one of four user files (MIEGRET, MIBATSE, MICOMPTTEL and MIOSSE), depending on who initiated the session. If the entry consists of data, bytes 4 through the end of the record are translated to EBCDIC and written to a dataset. If the entry is a message trailer, bytes 4 through 32 are translated to EBCDIC and written to the dataset. Finally, the dataset is closed and the event is queued to the monitor. These steps are repeated for each event until the queue is empty.

A client transaction is started by CSFI when a call packet is to be sent out to the GRO Instrumenter. The files to be sent are read from the X25CNT.DATA file using the READCNT transaction. Based on the destination, the proper connection is established using the CONN macro. A session initiation request packet is built and a session initiation request message is sent to the secondary partner. When a message is received from the secondary partner, the message is deciphered. If the message block received is a session initiation acceptance block, the client sends a message initiation request message and waits for the next message. If a message initiation acceptance message is received, the client sends the data packets. After the data packets are received by the secondary partner, the client sends a message termination request message and waits for the next message. If a termination confirmation message is received, the client sends a session termination request message and waits for the next message. If a session termination confirmation message is received, the session with the secondary partner is ended.

In addition to the CSFI software, the current GRO ICS uses the X.25 Network Control Program Packet Switching Interface (NPSI) software from NCR. NPSI provides an interface for Systems Network Architecture (SNA) users to use X.25 Packet Switched Data



Networks (PSDNs) in conjunction with their existing networks. This interface allows SNA host processors to communicate with both SNA and non-SNA equipment over PSDNs that use X.25 protocols. The NPSI software has General Access to X.25 Transport Extension (GATE) features that enable a host application program to completely control the establishment and clearing of X.25 virtual circuits. With this software, the GRO ICS is able to control the operation of the packet level protocol on individual virtual circuits, the operation of the X.25 interface for including incoming and outgoing calls, and the status of the link.

## **7. CONCLUSIONS**

The GRO ICS provides a vital electronic link between the Command Management Facility at GSFC and the remotely-located GRO Instrumenters. The successful design of the GRO ICS depends on the ISO-OSI 7-layer protocol model, X.25, the client/server concept, packet switching technology, and commercial CSFI and NPSI software. The GRO ICS was the first system in the United States to use the IBM CSFI software under the MVS operating system. The NCR NPSI software with GATE features was very helpful for connecting the SNA host with non-SNA hosts on the X.25 packet switched data network.

## **ACKNOWLEDGMENTS**

The authors would like to thank Patricia Lightfoot, Mike Paoletta and Alan Johns at NASA-GSFC/Code 514 and Ellen Stolarik, David Warren, Vijaya Sonti and Ronald Littlefield at AlliedSignal Technical Services Corporation for their continual support and many contributions to this communications task. This communications task is supported by NASA contract #NAS5-27772.

## REFERENCES

1. NASA/GSFC, Interface Control Document between the Packet Processor (PACOR) and the Command Management System and the PACOR and CMS Users, Goddard Space Flight Center, Greenbelt, MD, Nov. 1986.